IN THE CLAIMS:

1. (Currently Amended) A computer-implemented reduction checksum generator for calculating a checksum value for a block of data, comprising:

a reduction unit having a plurality of reduction stages and configured to pipeline a plurality of segments of said block of data through said plurality of reductions stages to reduce said plurality of segments to at least two segments; and

a checksum unit configured to generate a one's complement sum of said at least two segments and invert said one's complement sum to produce said checksum value.

2. (Original) The reduction checksum generator as recited in Claim 1 wherein said reduction checksum generator further comprises at least two registers, said reduction unit is further configured to iteratively reduce said block of data by storing said at least two segments in said at least two registers between iterations and using said at least two segments as part of said plurality of segments in a next iteration if said block of data contains a number of segments greater than said plurality of segments.

3. (Original) The reduction checksum generator as recited in Claim 1 wherein each of said plurality of reduction stages includes at least one reduction sub-unit, said at least one reduction sub-unit is configured to reduce three input segments to first and second output segments.

4. (Original) The reduction checksum generator as recited in Claim 3 wherein said at least one reduction sub-unit having full adders, each of said full adders configured to receive one bit from each of said three input segments associated with a same bit position.

5. (Original) The reduction checksum generator as recited in Claim 4 wherein said first output segment contains sum bits from said full adders, said second output segment contains carry outputs from said full adders bit shifted left and a most significant one of said carry outputs stored in

a least significant bit position of said second output segment.

6. (Original) The reduction checksum generator as recited in Claim 3 wherein said reduction unit is further configured to pass non-reduced segments from one of said plurality of reduction stages to another of said plurality of reduction stages if a total number of input segments for said one of said plurality of reduction stages is greater than a number of said input segments reduced by said plurality of reduction sub-units within said one of said plurality of reduction stages.

7. (Original) The reduction checksum generator as recited in Claim 1 wherein said plurality of reduction stages are configured to employ a leveled reduction or a 3-to-2 reduction to reduced said plurality of segments to said at least two segments.

8. (Currently Amended) A computer-readable storage medium having encoded thereon a computer readable program which, when combined with a computer, carries out the following method for calculating a checksum value using reduction for a block of data, comprising:

employing a plurality of reduction stages to reduce a plurality of segments of said block of data to at least two segments; and

generating a one's complement sum of said at least two segments, incrementing said one's complement sum if said one's complement sum generates a carry, and inverting said one's complement sum to produce said checksum value.

9. (Original) The method as recited in Claim 8 wherein said employing includes interatively reducing said block of data by saving said at least two segments between iterations and using said at least two segments as part of said plurality of segments in a next iteration if said block of data contains a number of segments greater that said plurality of segments.

10. (Original) The method as recited in Claim 8 wherein each of said plurality of reduction stages includes at least one reduction sub-unit, said at least one reduction sub-unit reduces

three input segments to first and second output segments.

11. (Original) The method as recited in Claim 10 wherein each of said at least one reduction sub-unit having full adders, each of said full adders receives one bit from each of said three input segments associated with a same bit position.

12. (Original) The method as recited in Claim 11 wherein said employing includes storing sum bits from said full adders in said first output segment, storing said carry outputs from said full adders in a bit shifted left position in said second output segment, and storing a most significant one of said carry outputs in a least significant bit position in said second output segment.

13. (Original) The method as recited in Claim 10 wherein said employing includes passing non-reduced segments from one of said plurality of reduction stages to another of said plurality of reduction stages if a total number of input segments for said one of said plurality of reduction stages is greater than a number of said input segments reduced by said plurality of reduction sub-units within said one of said plurality of reduction stages.

14. (Original) The method as recited in Claim 8 wherein said employing said plurality of reduction stages to reduce said plurality of segments includes employing a leveled reduction or a 3-to-2 reduction to reduced said plurality of segments to said at least two segments.

15. (Currently Amended) A computer-implemented parallel reduction checksum generator for calculating a checksum value for a block of data, comprising:

a plurality of reduction units having a plurality of reduction stages, each of said plurality of reduction units pipelines M segments of said block of data through said plurality of reduction stages to reduce said M segments to N segments;

a second level reduction unit having a plurality of second level reduction stages, said second level reduction unit pipelines said N segments from each of said plurality of reduction units through

said plurality of second level reduction stages to reduce said N segments from said each of said plurality of reduction units to first and second checksum segments; and

a checksum unit that generates a one's complement sum of said first and second checksum segments, increments said one's complement sum if said one's complement sum produces a carry, and inverts said one's complement sum to produce said checksum value.

16. (Original) The parallel reduction checksum generator as recited in Claim 15 wherein said parallel reduction checksum generator further comprises N registers for each of said plurality of reduction units, each of said reduction units iteratively reduce said block of data by storing said N segments in said N registers between iterations and using said N segments as part of said M segments in a next iteration if said block of data contains a number of segments greater than said M segments times a number of said plurality of reduction units.

17. (Original) The parallel reduction checksum generator as recited in Claim 15 wherein each of said plurality of reduction stages and each of said plurality of second level reduction stages include at lease one reduction sub-unit, said at least one reduction sub-unit is configured to reduce three input segments to first and second output segments.

18. (Original) The parallel reduction checksum generator as recited in Claim 17 wherein said at least one reduction sub-unit having full adders, each of said full adders configured to receive one bit from each of said three input segments associated with a same bit position.

19. (Original) The parallel reduction checksum generator as recited in Claim 18 wherein said first output segment contains sum bits from said full adders, said second output segment contains carry outputs from said full adders bit shifted left and a most significant one of said carry outputs being stored in a least significant bit position of said second output segment.

20. (Original) The parallel reduction checksum generator as recited in Claim 17 wherein

each of said plurality of reduction units pass non-reduced segments from one of said plurality of reduction stages to another of said plurality of reduction stages if a total number of input segments for said one of said plurality of reduction stages is greater than a number of said input segments reduced by said plurality of reduction sub-units within said one of said plurality of reduction stages.

21. (Original) The parallel reduction checksum generator as recited in Claim 15 wherein said plurality of reduction stages employ a leveled reduction or a 3-to-2 reduction to reduced said M segments to said N segments, and said plurality of second level reduction stages employ a leveled reduction or a 3-to-2 reduction to reduce said N segments from each of said plurality of reduction unit to said first and second checksum segments.